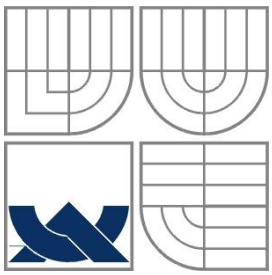
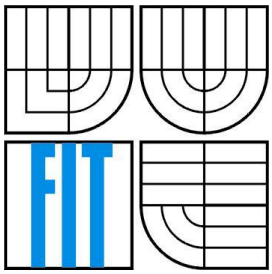


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

MIND MAP POZNÁMKOVÝ BLOK ONLINE

MIND MAP NOTEBOOK ON-LINE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR
JÁNOS GYÖRGY

VEDOUCÍ PRÁCE
SUPERVISOR
ING. SVOBODA PAVEL

BRNO 2012

Abstrakt

Myšlenková mapa je kreativním a nekonvenčním nástrojem, který svému uživateli slouží ke studování a objasňování myšlenek a souvislostí mezi nimi. Multimediální platformy jako jsou například Adobe Flash či Microsoft Silverlight umožňují přesunout aplikace z desktopových systémů na web a tímto je zbavit závislosti na jedné platformě a zpřístupnit k užívání široké základně uživatelů.

Tato práce se zabývá implementací online aplikace pro vytváření a spravování takovýchto myšlenkových map.

Abstract

Mind map is a creative, unconventional tool for taking notes and for clarifying ideas. Today's multimedia platforms, such as Adobe Flash, Microsoft Silverlight create a perfect background for integrating mind maps into on-line platforms.

This thesis represents the implementation of a web-based application for creating and managing mind maps.

Klíčová slova

myšlenková, mapa, online, Flash, Silverlight, poznámka, myšlenka, kreativní

Keywords

mind, map, on-line, Flash, Silverlight, note, idea, web-based, creative

Citace

János György: Mind map notebook on-line, bakalářská práce , Brno, FIT VUT v Brně, 2012

Mind map notebook on-line

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Pavla Svobody. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
János György
14.5.2012

Poděkování

Poděkování patří všem profesorům podílejících se na mé výuce po celou dobu mého studia na Fakultě informačních technologií VUT v Brně.

Contents

1 Introduction

Representing our thoughts with signs is a basic human behavior. We write down information for sharing, recording or learning them. It is important what we write, but also the way we are writing with.

When it comes to the area of note-taking the most important aspects are the efficiency and clarity. The process of putting the thoughts onto the paper is the result of how we imagine things, so if we want to find the most efficient way of representing the information, we have to understand how the human brain is functioning. Scientists are constantly exploring the construction and the operation of the brain, but there is still much to learn about. In the last couple of years many new discoveries have been made in this segment. Using these new findings in practice resulted in creation of mind maps, the main subject of this thesis. Most people are visually oriented. This means, that they are more receptive when they see a graphical representation of an idea. Mind maps offer a great alternative to organize information, all the important aspects could be seen at the same time, this could have a positive impact on the memory recall, learning takes less time and demands less effort. Creating and viewing them is intuitive.

This thesis describes the realization of an application for creating mind maps in web-based platform, bringing them closer to the ordinary users. The next paragraph contains information about the graphs, and the psychical aspects. These topics serve as the theoretical fundaments for the project. Then the third paragraph informs about the technologies, which were examined during the analysis, as well as the existing mind mapping tools. The fourth and the fifth paragraph contain the specification and the proposal of the project. These two parts clarify how the final product is going to look like, which functionalities will be deployed, and in what kind of form. The last two paragraphs cover the practical sides, the implementation and the testing. Important components and functions are described here, as well as the deployment and the evaluation of the project as a whole. The final product can be reached on the URL: http://www.stud.fit.vutbr.cz/~xgyorg00/IBP/.

2 Theoretical background

Mind maps are special instances of graphs. This capitol will give some background information about graph theory in general. It will list the graph types and the basic concepts in graph theory. The second part of this capitol is going to present the psychical aspects of mind maps, split-brain researches and radial thinking.

2.1 Graph theory

Graph theory should not be confused with the graph of a function or other terms used in computer science. Graph theory is a mathematical discipline, which uses manifold structures, to model connections between a set of items. In practice, graphs are used to model various systems in biochemistry, electrical engineering, computer science, or operational research and brought many results in the area of genetics research, computer network security or frequency assignment in mobile phone networks.

The origin of graph theory is connected with the name of a Swiss mathematician Leonhard Euler, who published a paper in 1736 called "The Seven Bridges of Königsberg" witch was named after a significant mathematical problem [1]. Königsberg is the former name of Kaliningrad, Russia, which is situated on both sides of the river Pregel, with two additional islands between the two branches of the river. The islands and the mainland were connected to each other with seven bridges. The problem was to find a path witch leads thru the city, and all bridges should be crossed only once, from one direction. Euler proved that there is no such path. The difficulty was the creation of a new technique for solving the problem. He pointed out, that the mass of the islands is not important in the solution, only the connection between them, so they can be represented with any kind of shape or symbol. Next, he realized, that when one leaves an island by a bridge, he enters another land by that bridge. In other words, during a move in the graph the number of entering the set of nodes equals the number of leaving them. This way each piece of land should be connected with an even number of bridges (except the one chosen as the starting- and the end-point), if we want to pass them only once. In this case it is not true, so this proves that the problem cannot be solved. Euler showed that the ability of crossing the road depends on its degree. The degree of a node is the number of the roads crossing it.

Generally a graph (marked as G) consists of two objects: nodes and edges. [2] An edge has two end-points in two nodes, other said it is connecting two nodes, and also can be represented with a pair of edges, or with an ordered pair, in case of a directed graph. The node is usually drawn as a circle or a dot, and the set is denoted as V(G). An edge is drawn like a straight or a curly line, connecting two end vertices. An edge that is drawn between the nodes A and B is denoted by AB, and E(G) denotes the set of edges. Under the term order of a graph, we mean the number of the nodes, it is signed as |V(G)|, and by size we mean the number of its edges. |E(G)|. An edge can be a link, if it is connecting two edges, or a loop, if it has the same starting, and end-point. We have different types of graphs. We say that a graph is simple, if there are no multiple-edges (more edges connecting the same nodes) or loops, multi-graph if multiple edges are present, or pseudo-graph if there are both multiple edges and loops. In order to differentiate between graphs with the same build, but different vertex and edge set, labeling is used. Graphs can be vertex-labeled and edge-labeled, according to the nature of the information it is represented with the set of objects. A hyper-edge is an edge, which can take on

more than two vertices, the graph, that allows such an edge is called a hyper-graph. Actually an edge is a special type of a hyper-edge, namely a two-uniform hyper-edge. Further standard mathematical terms can be used to describe a graph. Null graph is empty, and contains no vertexes. Infinite graph contains unlimited number on nodes.

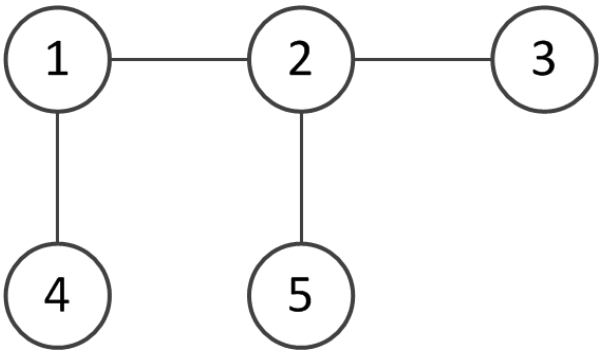


Figure 1: a drawing of a graph

2.2 Psychical background

In the middle of the twentieth century William Van Wagenen developed a radical surgical treatment for treating epileptic patients. He separated the two hemispheres of the brain to prevent the epileptic seizures[1]. A few years later Roger W. Sperry [2]started to work with a man who underwent this treatment. The separation of the two sides of his brain made them able to unravel the difference between the functioning and nature of the two halves. The right one excels in nonverbal and spatial tasks, while the left one is dominant at speaking, writing. Analogically, the left hemisphere is the logical, while the right is the intuitive one.[3] They also pointed out, that one of them is always more dominant, than the other one, thereby, everyone is either right hemisphere-oriented, or left hemisphere-oriented. Surveys assured that the majority of the population regardless of language or nationality is left-side oriented. This indicates that linear note-taking is the common used style, which has three major types, shown on Figure 2. Many people combine these three styles, but the main characteristics are the same. The main disadvantage of these styles is that important keywords often appear on different pages, obscured by mass of other, less important words. This prevents the brain from making appropriate associations between the key concepts. Linear notes also seem monotonous and boring. As such, they will be rejected and forgotten.

When we are taking notes, or try to solve a problem by sketching up ideas, mostly the left side of our brain is working. We can express ourselves, make logical connections between sequences of ideas, and sooner or later we will memorize the topic, or find the solution to the task, but what can we do to further improve our effectiveness? To get the most out of our brain we have to utilize our right hemisphere, use them alongside, and benefit from our whole mind. This is the main concept of mind maps. By forcing ourselves to visualize the connection between the particles of the topic, we unconsciously start to use a definitely bigger percent of our “computing power”, making associations

[1] Corpus callosotomy
[2] Autobiography of Roger W. Sperry
http://www.nobelprize.org/nobel_prizes/medicine/laureates/1981/sperry-autobio.html

faster and more intuitively. The human brain is extremely adaptive, that is why the more someone uses this technique, the better he gets in solving difficult tasks, learning, or in other activities benefiting from mind mapping.

DEFINITION:

„The Mind Map is an expression of radiant thinking and is therefore a natural function of the human mind. It is a powerful graphic technique which provides a universal key to unlocking the potential of the brain. The mind map can be applied to every aspect of life where improved learning and clearer thinking will enhance human performance." [5]

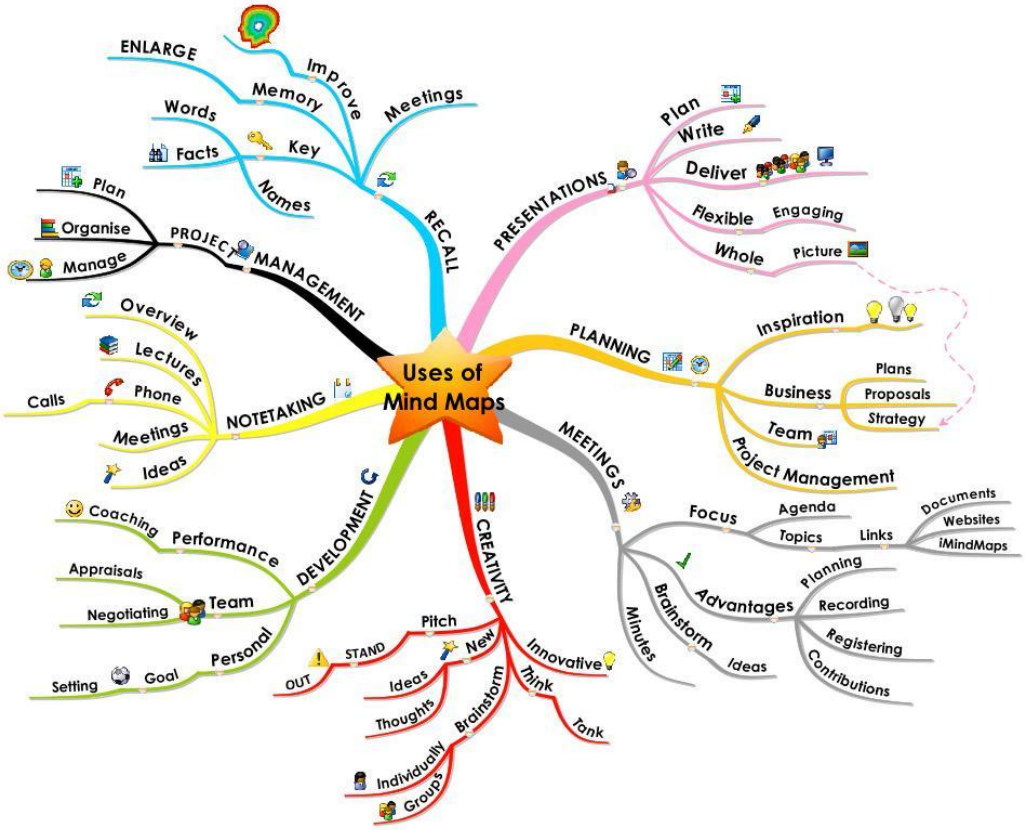


Figure 2: Mind map representing the use of mind maps themselves [6]

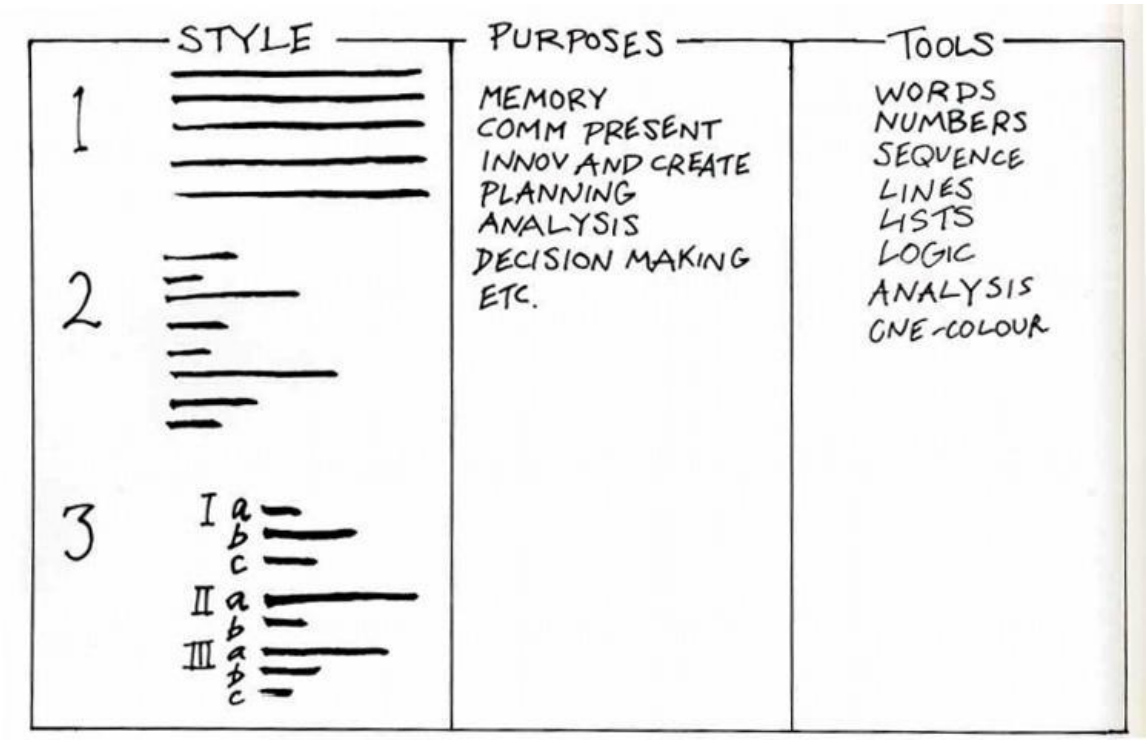


Figure 3 : Main note-taking styles:1-narrative style, 2-list style, 3-outline style [5]

3 Analysis

The application has been made using Adobe Flex. This part is going to give information about all the used technologies, and the reasons they have been chosen. There will be a comparison of the available programming languages, focused on their pros and cons. Next, it will inform about the existing editors fulfilling comparable tasks, and also they're analysis, in which areas are they performing good or poorly. This paragraph is going to clarify, that Adobe is providing appropriate tools for developing such an application.

3.1 Available technologies

The evolution of the web-based platforms is one of the most dynamical fields of today's software technology. As the internet is the biggest medium in our time, computer software companies are competing for the biggest share. As the importance of smartphones, tablets and other multimedia devices is rising, other factors are influencing the area as well. In this complex and fastidious environment, it is very hard to say which technology is better, or what the best choice is, if we want to create a long standing application.

From the appearance of Adobe Flash it has been the absolute market leader, and practically had no real competition. Then in 2007 Microsoft announced Silverlight, as an alternative, and announced it as "flash killer". The biggest advantage of Flash over the Silverlight was its maturity, but Microsoft managed to keep up by providing tools and features, that developers were missing from Flash. From the beginning Silverlight is searchable and able to index by searching engines, since the text representation is simple, and that was a huge advantage when business reasons came into calculation. Meanwhile the World Wide Web Consortium (W3C) started developing the next version of the internet's core technology, the HTML. They started with big ambitions, the HTML5 is intended to eliminate the promiscuity in the web environment, with providing multimedia and advanced application programming interfaces for web applications.

The new standard received intense publicity in 2010, due to Steve Jobs, who published a public letter "Thoughts on Flash" [7]. In his opinion, HTML5 is going to be the successor of Flash, and in a couple of years it will completely eliminate the need of a third party plugin for creating and displaying multimedia in the web-environment. As it turned out, he might had right, in November 2011, Adobe announced that he discontinues the Flash support for mobile devices [8], and will concentrate of developing tools to utilize the standard for the W3C.

Despite of the present trends, we cannot state that any of the current technologies is depreciated, or not suitable for further development. It means that the developer has to choose very carefully, and consider all the pros and cons each of them can provide.

3.1.1 Adobe Flash

Flash started as a simple animation-builder, but today it is a cross-platform browser-based application run-time that delivers videos, animations and interactive applications to web pages. It´s plugin, the Adobe Flash Player could be found on 96% percent of today's computers (according to Millward Brown survey, conducted July 2011 [10]).

Flash uses vector graphics, rather than rasters. It excels on larger screens, and provides high performance and quality in comparison with the raster graphic. It also has downsides: above a certain level of complexity it is not possible to represent all the shapes with formulas. The file size can be surprisingly small, due to the high level of compression, what the SWF (Shock Wave Flash) format provides. The reduced size comes with a price, essentially crawlers and search bots are not able to read and index the content, and other solutions have to be used in order to that. The video is implemented through FLV file format and it supports manifold video and audio compression standards, including H.264, MP4, MP3, AAC. [11] ActionScript is used to program Flash objects. It is a programming language based on ECMAScript, what is very similar to the commonly used JavaScript. It can be integrated with PHP, ASP and Ruby on Rails. It has an enormous class library for developers to make manifold online and stand-alone desktop applications. More versions are present, but the latest is the object oriented ActionScript 3.0. It can execute ActionScript up to ten times faster then it´s predecessors and adds some hardware acceleration capability (OpenGL, DirectX). [12]

3.1.2 Microsoft Silverlight

As I previously mentioned, Silverlight was created to be a strong competitor for Flash, although the first versions were focused on media streaming. In the latest versions Microsoft equipped it with all kinds of multimedia support, as well as developer tools and support for CLI languages.

Silverlight uses different strategies on many fields in comparison with the Adobe Flash. Primarily, it uses WPF animation model, which is time-based, so there is no need to specify the content of each frame. Uses an uncompressed data format called XAML, Scripting is implemented through more languages including Visual C#.Net and Visual Basic.Net. Media encoding support is from ubiquity reasons similar to the ones Flash is supporting, adding some further standards. The areas Silverlight excels is generally the scripting, animation, video and audio capabilities, the places, where it falls short is the adoption.

3.1.3 HTML5

It is very important to stress, that HTML5 is still under development, and cannot consider as a stable and supported standard. The forecasts are very positive about it, and as I already mentioned , many professionals consider it as the „future of the web", but with any other new technology, it will take time to fully utilize and adopt. The HTML5 is in the most of the cases mentioned in connection with the other two technologies above. Because of this many people think it is using the same strategies to achieve interactivity as the others, but that is not true.

It is not just a replacement for a plug-in, it makes the web a development environment, by empowering it with JavaScript, to be able to create applications. [13] This new standard is the result of a massive unification by the W3C, it contains high amount of new specifications. Basically it is built up from three parts:

- HTML
- Cascading Style Sheet (CSS)
- Javascript

Each part is responsible for a specific behavior, and the majority of the overlaps had been eliminated, for example presentational elements such as <font> or were dropped, these attributes can be set only through CSS for now on. Multimedia elements are inserted using <video> and <audio>.

3.2 Other used technologies

Creating the application demanded the use of other software development tools and languages. The realization greatly benefited from the tools Adobe is providing for developers, mainly the Flex platform, that they brought out for creating enterprise level applications. PHP and MySQL has to be mentioned too, as they were used them to implement the database-side of the application.

3.2.1 Adobe Flex

Flex is a programming framework developed by Adobe Systems for developing Rich Internet Applications (RIAs).

A flex application uses a custom XML namespace called MXML as the basis of the code. This markup is easy to navigate and to see-through. The development in this environment is component based and it has an impressive component library. Interactivity not covered by native Flex components is achieved by using ActionScript, like in any Flash application.

Adobe Flash Builder, originally entitled Adobe Flex Builder is an Eclipse based IDE for developing Adobe Flex applications. It has a dual view:

- Design: WYSIWYG (What You See Is What You Get), The language components can be pasted to the canvas using an user interface This is useful to make the application as aesthetically pleasing as possible.
- Code: Markup can be written by hand, it provides an impressive code assist, what can make the development definitely smarter.

The components are highly customizable, they can be skinned using CSS, but PNGs, JPGs can be also used. Adobe Flash Builder features the ability to monitor all network data between the client end application and the server back end. This allows developers to fully debug the application and streamline the application's network usage. They can choose between the Builder, or can even import Flex SDK/Compiler into Eclipse for free.

3.2.2 FlexLib

FlexLib[3] is a community project, with the aim to create open source GUI components for Adobe Flex. The library contains high quality components, where each is derived from original Flex components, with additional functionality or improved behavior.

3.2.3 PHP: Hypertext Preprocessor

PHP (recursive acronym for PHP: Hypertext Preprocessor) is a free to use scripting language, that can be embedded into HTML, which makes it perfect to use for web development. It is a so-called server-side language, what means, that scripts are performed on the side of the server, and the user gets only the interpreted output.

The syntax of PHP is originated from more scripting languages, originally from Perl and from C. PHP contains a great dozen of libraries for working with all kinds of databases, strings, matrices, and for using network-protocols such as HTTP, SMTP. Nowadays it is frequently used in combination with database systems and web servers (Apache).

3.2.4 MySQL

MySQL is an SQL-based database management system, which can handle multiple users at the same time. MySQL is the most popular database manager, thanks to the fully open LAMP (Linux-Apache-MySQL-PHP) compilation, which can provide a low-cost, simple-to-calibrate dynamic web-service.

MySQL is an appropriate tool for storing user passwords, it has built-in functions, such as the PASSWORD() for encryption, or even better is to use hashing algorithms, for example SHA1 or MD5. Unfortunately nowadays exploits to these functions are quite common [15], but it is still better to use them for encryption, than store the passwords as plain text.

[3] FlexLib website: http://code.google.com/p/flexlib/

3.3 Existing mind-mapping tools

Teaching experts and software developers have already recognized the upsides of mind maps, and also the potential of integrating them into the web-environment. Several applications were made, this paragraph is going to introduce the most relevant ones, which also resemble the functionality what the project was aiming to achieve. During the research I was concentrating on the appearance as well as the functions included, trying to think of new features that has never been implemented before. At each application I tried to remain objective, but I also include my personal opinion. When testing performance, I used more browsers, and different computer configurations with different computing power.

3.3.1 Mindomo[4]

In 2007 Mindomo started out as a simple mind mapping application, entirely for free offering basic functions, like creating saving and extracting maps. If someone visits Mindomo's website, can now find and extensive and powerful mind mapping application offering free and premium features as well.

As initially, it is an Adobe Flash application, with a downloadable version running on Adobe Air platform. It has an attractive user interface, presenting in decent colors. The control is intuitive, the users can pick up quite easily and on the working plane they are no bothering elements, the map is in the center of the attention. For customizing the nodes, one can use the upper panel, or a separate panel can be opened by clicking onto the node. The view controls are on the bottom, they give the user the ability to zoom in or out, center the screen and search through the map. These functions come handy when we are working with an extensive map, but the zoom effect itself is slow and clumsy. The nodes are highly customizable, and more types of multimedia elements can be attached to them, including picture inserting through an URL. It appeals on team work, users can share they work, favorite maps. The places it falls short, is the performance due to the complexity, on the netbook I experienced, that the movement is sometimes cloggy. The overall impression was good, the application is easy and entertaining to use.

3.3.2 MindMeister[5]

MindMeister has been founded in 2007 by a German company. They're product has been a cloud application form the beginning, and has been sold as a SaaS (Software as a Service). As well as the previous one it went through massive development as the years passed, and now it is one of the most compelling mind-mapping applications on the market. They are offering a basic account for free, as well as more price plans with advanced features. Unfortunately, the basic plan has a lot of restrictions: the user can create only three maps, so it is practically good only for a test drive.

[4] http://www.mindomo.com/
[5] http://www.mindmeister.com/

MindMeister is entirely JavaScript/AJAX based, this means there is no need for second party add-on, if the used browser has JavaScript capabilities, then it is good to go. The technology also has its downsides, I experienced several performance issues with all the computers, I used for testing.

After the previous one, the user interface looks a bit crowded, with more widgets right on the screen, and the color of the controlling elements can be distracting. It can be seen that in the process of piling up the features, the performance, and the usability diminished. From examining this application it could be clearly seen, that Flash has much better capabilities in both animation and performance.

3.3.3 Bubbl.us[6]

I can tell in advance, that I found Bubbl.us the best mind-mapping tool on the market, or at least the best for the functionalities, I was looking for. In comparison with the other two, it is completely free, without any advertisements. It was founded in 2008, but since then, they kept it simple and sleek. There are no annoying elements, no complicated process flows, and it is responsive, in many scenarios almost instantaneous. Of course, it lacks features like project management, but in my case, this was not the aim.

The concept of functioning differs from the other two: the editing panel can be brought up with mouse hover, but also contains important tools, like zooming or basic node editing tools. The thing I found inferior, is the movement with the graph, what can be done through clicking on a separate button, and the colors of the user interface could be more elegant, esthetically pleasing.
Bubbl.us convinced me, that Flash is the technology, I am looking for, as it is a Flash application, just like Mindomo. Its built-up influenced me the most, and showed me, what are the basic functionalities, a mind-mapping application should handle.

[6] https://bubbl.us/

Figure 4 : Mind map created with Mindomo

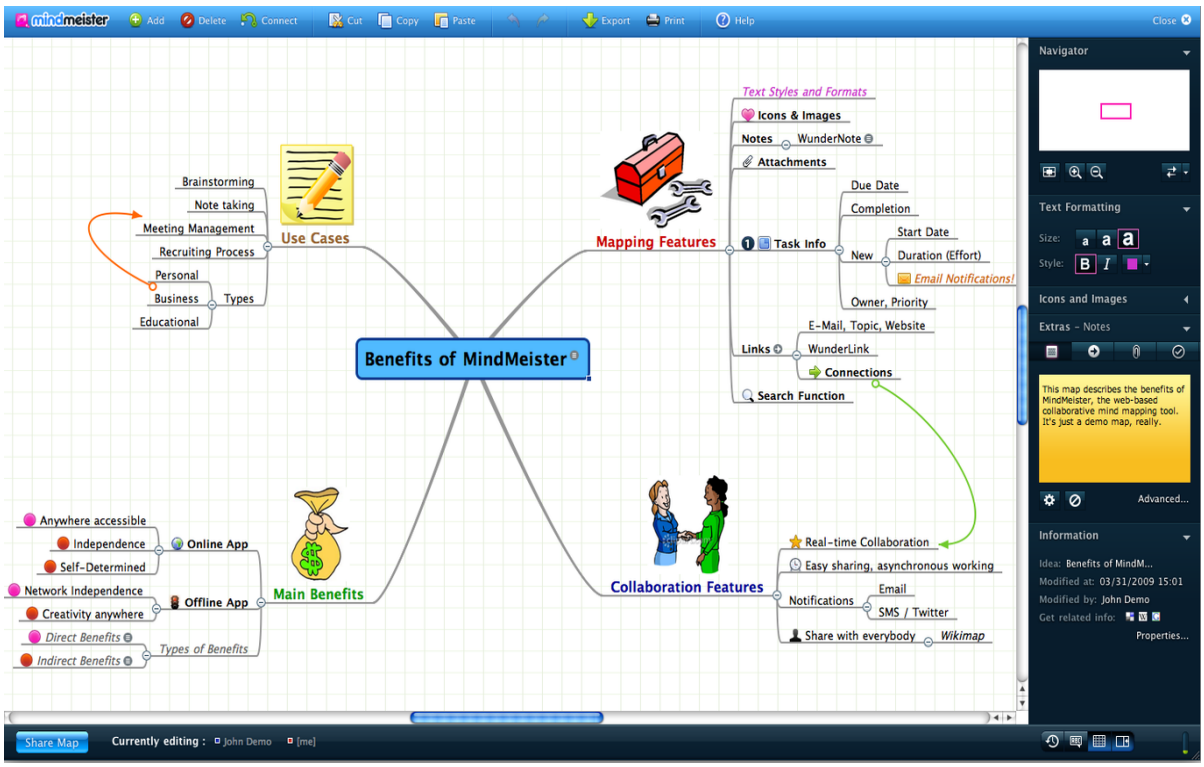


Figure 5 : Mind map created with MindMeister

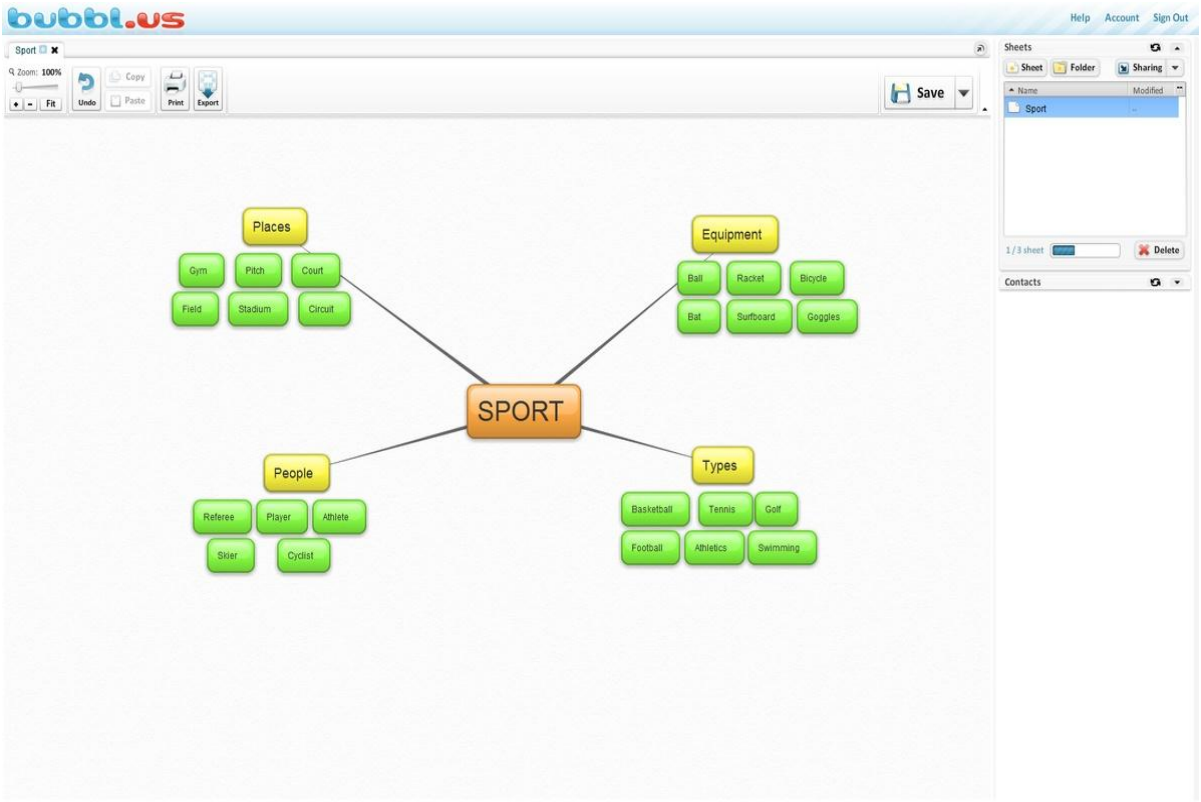


Figure 6 : Mind map created with bubbl.us

4 Specification

The task has not specified any strong requirements according to the user interface neither the functionality of the project. These details were discussed with the supervisor of the project, and during the first semester project defense. It was allowed to use any tools to achieve the expected outcome. This paragraph is dedicated to the functions and qualities, the application is capable of.

4.1 Global features

This thesis belongs to the category user interfaces. According to this, besides the usability, the emphasis was given to the esthetics as well. When we work up a topic, which is not commonly known, the user interface has to be simple, intuitive and easy to use. Without that, our application will be rather offensive and will not catch the attention. [17] The following points will inform about the standard features that are essential for a mind mapping application.

4.1.1 Users, map access

One of the most important features is the ability for handling multiple users. Users are going to be able to create, save, and manipulate they're own maps. The maps can contain personal or even confidential data, so they should be accessible only by the users, which are granted. This feature involves registration, password management, log-in and log-out functions.

All the maps owned by the user have to be available and visible at all times to give an insight. This means, that there should be a panel dedicated for the list of the maps already created. When there is a change in the files (file added or removed) this list has to refresh itself, to give the user instant feedback.

4.1.2 Portability

Almost every application, which has been investigated during the analysis, was given the ability to export the maps to a separate file. This makes them portable: the user can download the data file of each map, edit them manually, and re-construct them by opening it again. Portability makes also possible the sharing of the maps. Another way to achieve sharing would be the creation and management of member groups.

4.1.3 Graph customization

The most important component of the application is the graph, it has to be in the center of the attention at all times. To make the map attractive and easy to navigate, the nodes have to be customized in many ways. The things that have to be changeable: color, shape, size, unique position. Of course these changes should be permanent, as the node properties are saved with the map information. When speaking of the node-customization panel, we can find different solutions in different applications, but the important part is, that it has to be clear which node is edited at the time.

4.1.4 View

The application should give almost the same freedom of arranging the nodes as on paper, or even more. By this I mean that the nodes can be rearranged among the view space, and the distance of the viewpoint can be changed. This is quite essential when working with bigger maps in order to fit them into the screen. It is also important to have a button, which resets the view, in case that the map gets misplaced.

4.1.5 Compatibility

It is commonly known, that different web browsers behave differently in various scenarios. The aim is to make the application as widely adaptable- as it can be. In sight of this, the application has to be fully functional in the most used browsers (Mozilla Firefox, Microsoft Internet Explorer, Google Chrome). Since we are talking about a Flash application, the browser has to be equipped with Flash Player.

4.2 Additional functions

After setting the ground rules, the emphasis was given to the features that are going to make the application unique, what cannot be found in any of the already existing solutions. Of course, those previously mentioned applications are in development for years. Compared to them, the application has its limitations, but can fulfill a similar purpose, and can provide a comparable user-experience, and additional features.

4.2.1 Spring-like movement

As it was mentioned above, the look, and the overall feeling is important. In order to improve that, the movement with the mind map was implemented to look like a living creature. The official term for such behavior is “Force-based graph drawing”, because there will be imaginary forces between the

nodes, making them attract or distract each other. In result, it will give a very attractive look, and will distinguish the application from the others. On the other hand it will not able to remember the last positions of the nodes, but this is a small price to pay for such a feature.

4.2.2 Advanced notepad

Mind mapping is usually used for taking short notes, to sketch the idea and to make it more likely to remember. What has been added to this concept is that every node behaves like a separate text file. This way, the user can store not only short words, but can add sentences, or even a whole article to a specific thought. Additionally, with the integration of a notepad component, the writing experience is even better, styles and colors can be applied, just like in a regular document editing application.

4.2.3 Tabbed navigation

Using tabs is a common trend in internet browsers, but also in other applications, such as text-editors. This way of arranging is very effective, can clarify the working space. The application is using the same approach with the help of a tab component.

5 Proposal

5.1 Graphic user interface

As it was previously mentioned, Adobe provides Adobe Flash Builder that makes it possible to programmatically plan, and skin the components. The Adobe team certainty made big efforts to make the flex components flexible and attractive. There are several built-in skins, but they can be skinned manually as well. The skin named “Spark” is used in the project. In the followings I will go through the elements of the application, and give all the important details about them.

5.1.1 Welcome screen

When the application is loaded, the first thing that shows up is the welcome screen. It presents a mind-map themed background and an accordion on the right side, which serves more purposes. It can show the log-in dialog for existing users, in this case the user can access the application in the standard way. Next, the accordion can be switched to show the register dialog for new users. In this mode, the guest can create his/hers own account –again- the usual way. The only additional field is the one for the email address, which is verified with a single regular expression. This information is needed, so that the users could be contacted, if any administrational deeds justify that. The last tab on the accordion is for displaying basic information about the author and the application.

5.1.2 User panel

After signing in, the accordion is replaced by the user panel, and the log out button becomes available. The panel holds the name of the current user, a list of all the maps created in the system by the given user, and on the bottom there are four buttons for list-related functions. Maps can be opened either with the open button, or with a double click. The second button gives command to download the file; the third is for deleting it. The reload of the list is automatic, if anything list-related happens in the system, but a fourth button is added for reloading the map. This can be used, if anything unusual happens in the system, or to confirm the present state of the list.

5.1.3 Map area

The center region is reserved for the map itself. The display is limited for this area only. As the movement with the map object is free, it can happen that the user drags a part of it outside these boundaries. If the map is reaching beyond that, then it gets truncated. For node customizations, there is no solid panel. Instead, a floating panel is showed, when the user clicks on one of the nodes. This

solution is used to keep the user interface as simple as possible, showing these functions just when desired. Clicking on a specific option performs the task, or brings up additional panels.

- The user can add a new sub-node to any existing node. When the new node appears, it does not inherit the attributes of the parent node. It initialized to the following attributes: color: gray, shape: ellipse, name: "EMPTY".
- When clicking on the remove button, the node is deleted from the map instantly, and if the node has further sub-nodes, they are deleted as well. The root node cannot be removed, in this case the delete button is disabled.
- When the user chooses the repaint option, a color picker shows up. It offers colors from the palette, but the color can also be set with entering it in hexadecimal form. This is especially useful, when the user wants to set a series of nodes to the same custom color.
- The button for changing the shape brings up a second panel, where the user can choose between six shapes: circle, ellipse, rectangle, rounded rectangle, square and rounded square. The reason why these geometrical figures were chosen is, that they can be easily resized, when the name of the node is too long, and it would overfill the shape. A big star or a triangle would rather look unsightly.
- The next option is to lock the node location. When this feature is activated, a small black square appears at the top of the node. This indicates, that no matter how hard the graph is pulled, the nodes with locked position will not move. The motion can be reactivated by clicking on the lock button again.
- The last item in the Node Controls panel is the rename button. It brings up a dialog box, where the user can set the new name.

When the specific task is performed, the floating panel is closed automatically, but it can be closed by clicking next to it, anywhere on the screen.

As the previous paragraph described, every node is holding additional text content. For helping the user review these contents, there is a tooltip implemented. This tooltip shows a fragment of the text, which can be a big help in navigation.

5.1.4 Control panel and the tab bar

Above the map area is the panel with the file-related functions, and the bar for the tabbed navigations. The buttons placed here are essential, and cannot be placed anywhere else, because they are manipulating with the map as a whole:

- New Map: this button creates a new map. When the map is created, the core node is placed in the middle of the map area, and the initial inner structure is created for it.
- Save Map: does what it says, when this button is pressed, the changes made to the current visible object are saved on the server-side.
- Open Map: this button can be used to open previously created and downloaded maps. When pressing it, a file browser is showed and the file can be selected in the standard way.
- Zoom Controls: it has been already discussed, why are these controls necessary. The zooming is performed by levels, no effect is applied, since it unnecessary prolongs the zooming task.
- Center Map: helps navigation, by resetting the map coordinates. This can be helpful, when it gets misplaced. This task is also performed automatically when zooming, since the situation gets changed, and the placement has to be refreshed.

The tabs are showed above the control bar. When choosing the tab handler, there were several regards towards the tabs:

- Reorderable tabs, with the dragging of them
- Closeable tabs
- Scrolling tab bar if too many tabs are open
- Drop-down list of tabs
- The tab with the mind map should always be on the first place, and cannot be closed

5.2 Force directed graph drawing

As I have already mentioned in the specification, when moving the mind map, it acts like if the nodes were connected with strings. This movement is achieved by applying a force based algorithm. The majority of the layout-calculating algorithms are domain-specific. That means that they get information not just from the graph, but from the environment. Force-directed algorithms (also known as spring embedding) on the other hand are using only information stored within the graph object. Graphs drawn with this method are symmetrical, and visually pleasing.

This layout could be imagined as the structure of the atoms in a molecule. There are two kinds of forces effecting between the particles:

- Repulsive force between each particle
- Attracting force between the particles, which are connected with some kind of nuclear bond

When these two forces extinct each other, the molecule is stable, and the atoms are standing still.
This settlement has been introduced by Fruchterman and Reingold in 1991 as „even vertex distribution“ [18]. The attractive and repulsive forces are defined to the followings:

$$F_a(d) = \frac{d^2}{k} \qquad F_{r(d)} = -\frac{k^2}{d}$$

Where d is the distance between the two nodes, and k is the optimal distance between them, k is defined as:

$$k = C\sqrt{area/2}$$

At first, I tried to use the Spring Graph Flex Component, which is developed by Mark Shepherd, and offers solution for creating organic-looking graphs. Although I managed to get it work, I was not satisfied with the behavior of the graph, as it is designed for bigger databases, and decided to create a my own algorithm. I have used the „Fruchterman and Reingold“version of the algorithm According to [9] this type can handle graphs up to fifty nodes without problem. In the practice a mind map is rarely larger than thirty nodes. This implementation of the force based graph drawing is using a special handler, the *temperature* variable. It controls how big the displacement can maximally be, its value is decreased as the graph reaches a better, a more stable layout. In order to determine whether the layout is better than before, the algorithm has to summarize the force affecting the nodes.

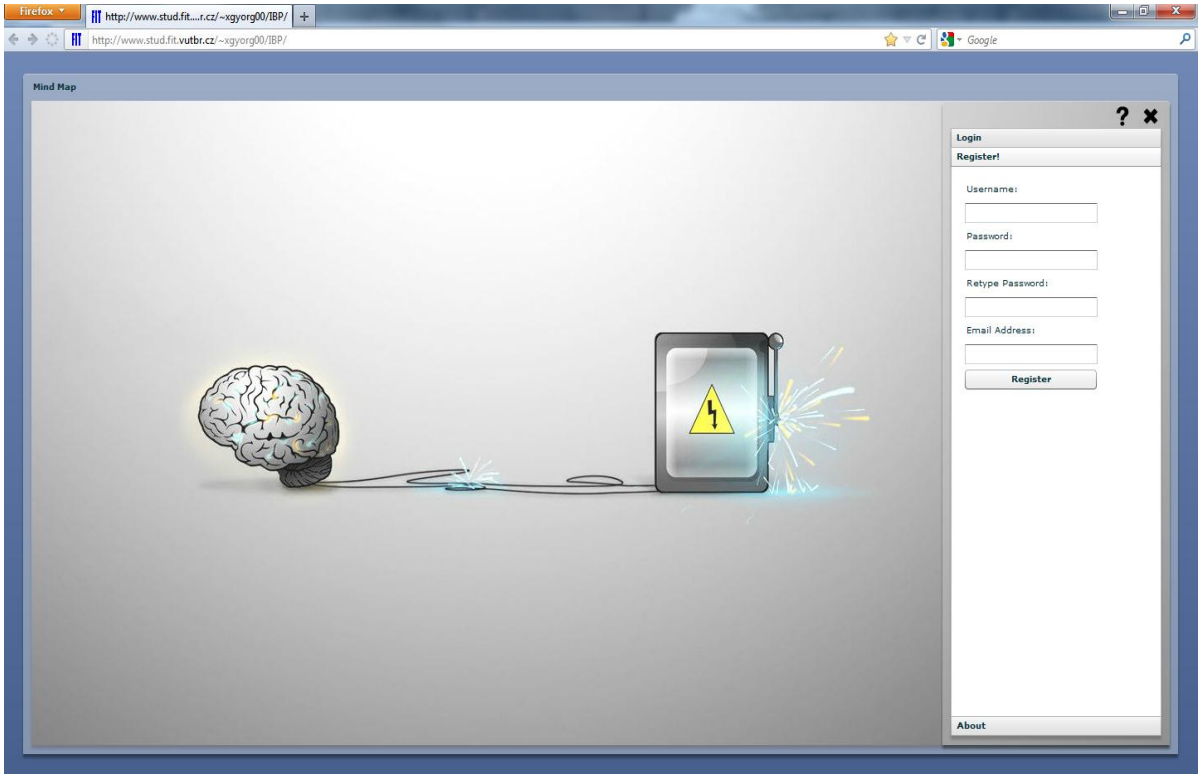


Figure 7 : Welcome screen

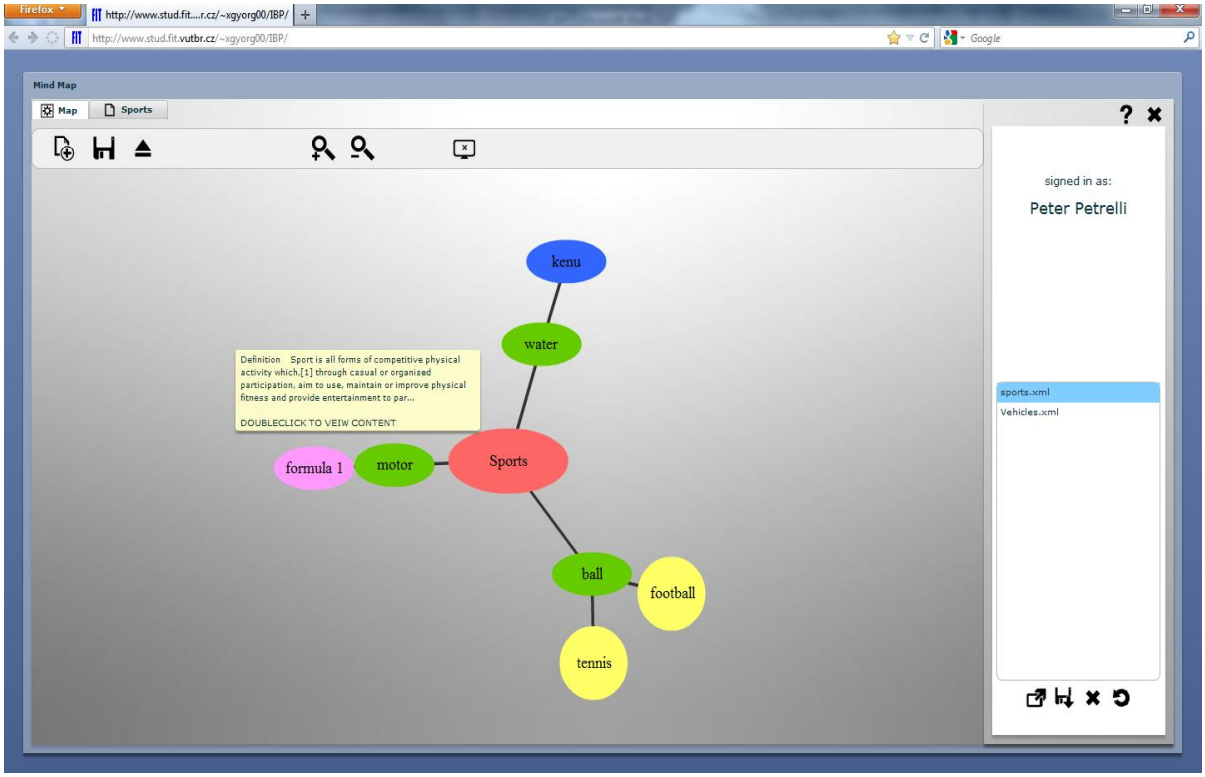


Figure 8 : Graph area

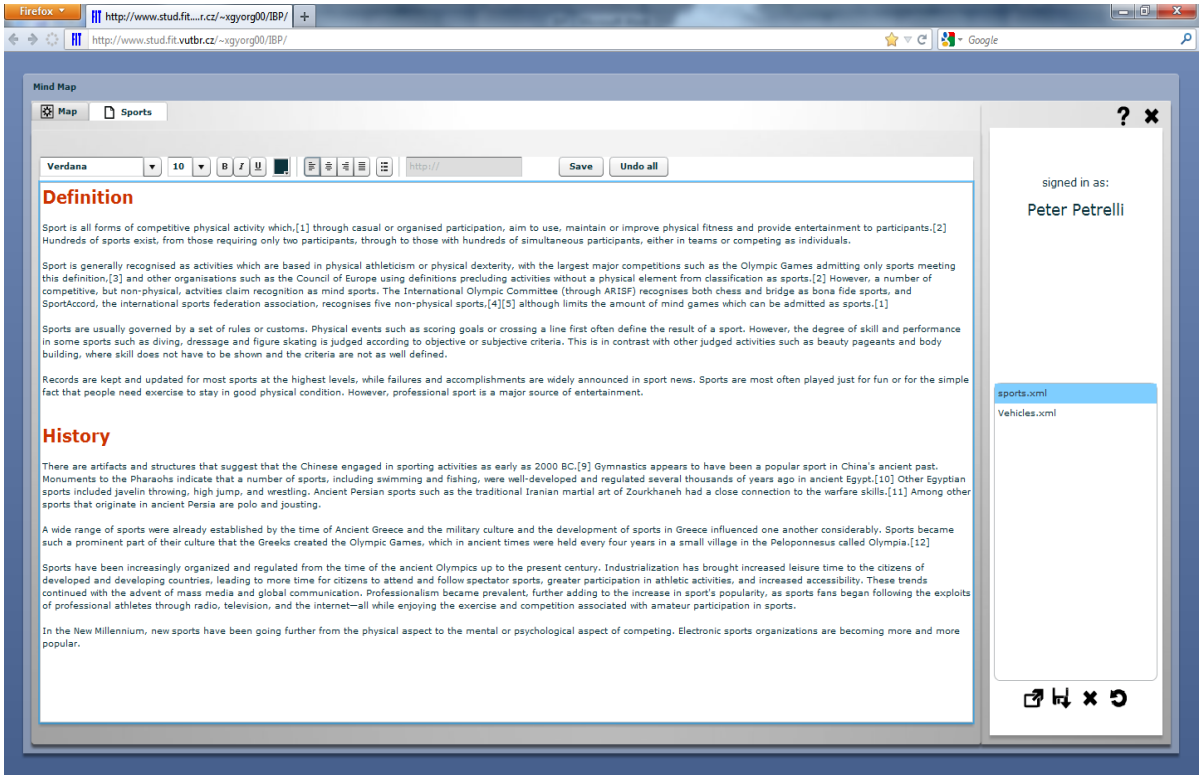


Figure 9 : Text area

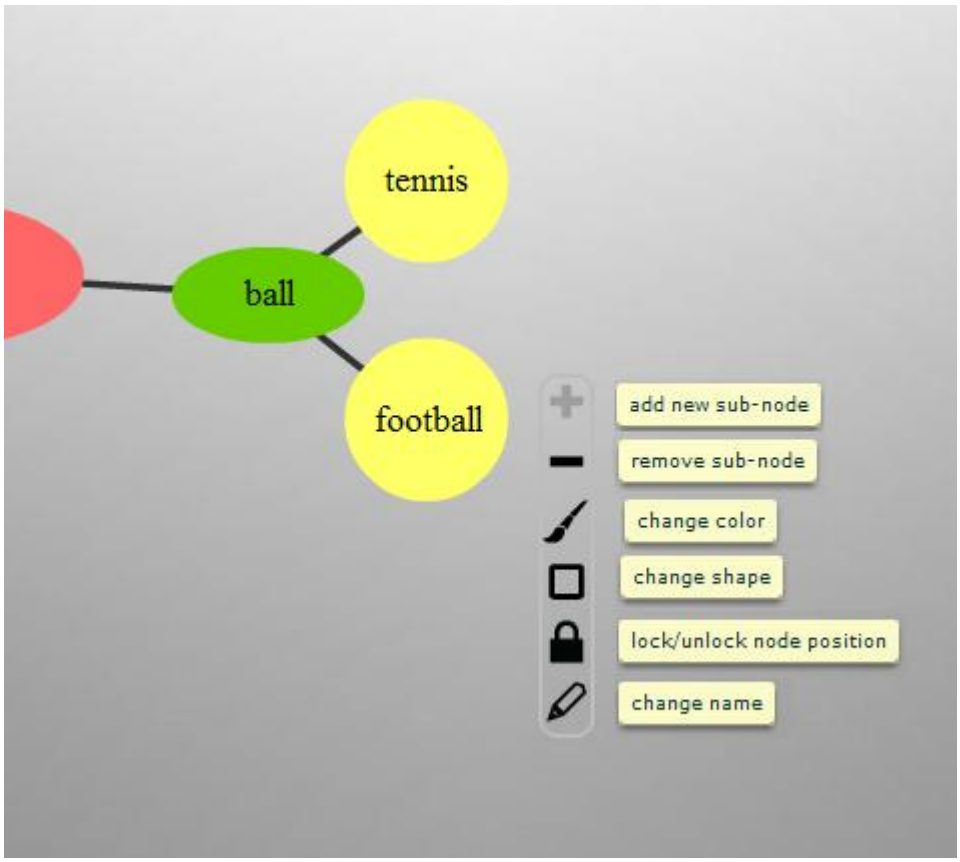


Figure 10 : Node controls panel

5.3 User database

Storing the user information has always been a very sensitive task. The creator is responsible for the confidentiality of the data stored within the system including user names, passwords, email addresses. Because of that, we have to be cautious, how to store the users data.

Storing the user data in XML would have its upsides, such as good performance, smart backups, no database server needed, but falls short when security comes into the picture. Using MySQL for the user database is a much better choice: it is more secure and has built-in parsing functions. On the other hand it requires support from the host, but nowadays MySQL support is standard at any web hosting service, even at the free ones. For securing the passwords it is sufficient to use cryptographic function MD5, which has built-in functions in MySQL and brings basic security.

5.4 Map database

When dealing with map storage, other aspects are important. The log-in information are needed just once - when the user steps into his/her account - but the manipulation of the map data is more frequent, and bigger amounts of information are processed. As it was previously mentioned, for this purpose, it is convenient to use the eXtensible Markup Language.

When dealing with the database structure, the Force Directed Graph Drawing Algorithm had to be examined that is responsible for the display. From the pseudo-code it is clearly visible, that there are two kinds of entities to store, the nodes, and the connections between them. This suggests storing the edges explicitly, in a separate area, where they can be easily accessed. The data structure looks like the following:

```
<map>
<Nodes>
<Node num="0" name="rootNode"/>
<Node num="1" name="firstChild" />
<Node num="2" name="13Child" />
<Node num="3" name="secondChild" />
<Node num="4" name="24Child" />
<Nodes/>
<Connection>
<Edge fromID="0" toID="1" />
<Edge fromID="0" toID="2" />
<Edge fromID="1" toID="3" />
<Edge fromID="2" toID="4" />
<Connection/>
</map>
```

6 Implementation

This capitol is going to summarize the implementation of the project. Important classes and functions are going to be reviewed in the followings. The application has been created using Adobe Flex Suite, that is used for coding enterprise level applications in Flash environment.

6.1 Node

A node is the elementary part of a Graph. It extends the built-in `Sprite` object, making it able to hold graphic elements. It consists of two main parts, the shape (one of the six already mentioned ones), and a `Label`, with the name of the node. The width of the shape is adjusted according to the length of the name, so it will not overflow.

As the force directed algorithm specifies, the node has two vector-type variables, which contains the force influencing the node and the velocity. ActionScript also has a pre-defined `Point` class, what is perfect for that purpose, it stores two integer-type variables, they can be attained by referencing them as *x* and *y*. The most important function of the object is the one, which redraws the entity. This function is called the most times during the run of the application, so it has to be as fast as possible.

This class is practically a `Sprite`, so its location can be easily set by adjusting its coordinates (x, y), and also easy to use resizing mechanisms built into ActionScript (`ScaleX`, `ScaleY`).

6.2 Graph

The Graph is the main object of the application. It extends the `UIComponent`, as it is holding the graphical representation of the visualized mind map, but also implements the interface for the manipulation with it. The visual part is stored within two arrays: the first is for storing the nodes, and the second for the connections between them.

When we want to open a map from the list, we only know the name, and the URL of the map. As I mentioned above, the graph gives command to the parser, and waits for the `XmlParsed` event to be dispatched. After that, the nodes are equipped with the appropriate event listeners:

- `MOUSE_OVER`: display the node tool-tip
- `MOUSE_OUT`: hide the node tool-tip
- `MOUSE_DOWN`: display the floating menu for node manipulations, or if the `MOUSE_UP` event does not arrive in a short period of time, it indicates that the user is holding the mouse button, and dragging is started.
- `MOUSE_UP`: drop the graph if it was dragged
- `DOUBLE_CLICK`: open the contents of the node in a new tab

- `ENTER_FRAME`: redraw the map according to the spring directed layout algorithm

The enter frame means that the associated function is executed on every frame, meaning that we get a fluent repositioning mechanism. Using the `ENTER_FRAME` has some flaws: if the function what it triggers is too complex, resource demanding, the movement can become slow or cloggy. To avoid this, we have to optimize the function as much as possible.

To further increase the efficiency, the graph is reframed only if it is appropriate, thus if the layout has once settled, it stops counting the overall force of the system, but only checks if the value is above a certain level. This value is manually increased above the level, when the graph is moved, or a node is added, or someone else justifies the repositioning. The concrete numerical value was ascertained by experimenting and using the trace function to track the variables.

6.3 XML Parser

The maps are initially stored in an XML file. When a map gets opened, or created, the application has to parse the data file, and create the object from it. This is processed by the `MapXML` class.
If we want to work with XML files in ActionScript, we have to use the `XMLDocument` class for instantiating them, and `UrlLoader` class for loading. Our files are located on a web server, so accessing them will be an asynchronous operation. Before calling the `UrlLoader.load` command, the `Event.COMPLETE` event property has to be assigned to the `UrlLoader` instance. When the application issues the command to reach for the XML file, it should wait until the data is delivered, and ready to use.

An `XMLDocument` consists of a root node, and an array of one or more child nodes. Each node can have further child, stored in nested arrays. There are two types of nodes: tag-based or text-based. This information can be obtained from the `Nodetype` property. The child nodes can be accessed with the `firstChild` and the `childNodes` commands, they can return the first child node or the array of child nodes. With these operations, the XML entries are organized into nodes, and returned to the Graph object.

The asynchrony caused a general problem. It could be solved by `extending` the `EventDispacher` class, this means that the parser class can dispatch custom messages, specifically the event `XmlParsed`, when the data processing is ready, and the data is prepared for further use.

6.4 Node Controls

This class implements the menu that is showed, when the user selects one of the nodes. It is holding the buttons for customizing them. When implementing the class I had to choose between two concepts:

- Every node owns a separate node control menu. This way the event listeners can be assigned directly to the nodes. This would increase the integrity of the architecture, eliminates the need

for more public functions. Also brings up problems, such as how to detect, that some other event has occurred, and the panel should be hidden.

- The graph has only one control menu, the concrete node is identified with the event.target item of the `Mouse.Event` instance. This way the Graph is implementing the functions, and then relays the action to the node. This is not as elegant as the previous concept, but saves a lot of memory, because the menu is not instantiated every time a node is added to the graph.

Both methods are correct, but there is sensible drawback of performance with the first method. Because of that, the class was implemented with the second concept.

6.5 Map list

According to the specification, the user has to have his/her maps available in the form of a list. Listing a directory with Flex is fairly easy, the `flash.filesystem.File` object has built-in functions for that purpose, but unfortunately it works only in Flash Air. In the Adobe Flash Player environment it is not allowed to use this object due to security considerations, so other tools had to be used. Again, as in the previous section, PHP has been used to settle the problem. The task could be simplified by creating a separate folder for every user. The mechanism is the following:

- The application makes an `HTTPService` object, fills it with the location of the user folder (fixed map folder `url` + `user name`), and executing the call using PHP. A result handler is deployed to listen for the answer.
- The PHP script gathers the names of the files, by collecting them into a string, divided with a separator, when the file processing is ready, the result is returned to the application. The mechanism collects all the files, no matter what their extension is, if the contents are correct according to the syntax, the application can handle it, and create a graph object from it.
- The returned data is turned into a `data provider`, which fills the list with the desired set of file names.

6.6 View states

The Flex environment provides a very useful tool, the possibility to make so called „ view states“. With the view states, the application behaves as it had more interfaces and the change between them is very easy. It is not necessary to redraw the whole region, if the user logs in, or out, it is enough to change the state. Setting the initial state, as well as changing it to another transpires through the “`currentState`“ command. Apparently, here are two main states.

The first state is the screen, which welcomes the visitors and it contains some general description about the application, the forms for logging in, and for the registration. In my case, I used the Accordion Flex component, so the users can easily change between these three options.

6.7 SuperTabNavigator

The `SuperTabNavigator` is an extension of the `TabNavigator` navigation container. It is located in the „flexlib“ library. It implements all the functions of a `TabNavigator`, but adds some further functionality:

- Draggable, re-orderable tabs,
- Closeable tabs,
- Scrolling tab bar if too many tabs are open,
- Drop-down list of tabs

That makes it a perfect component for using it for showing the contents of the nodes. Several texts can be opened at the same time.

After opening the contents of a node, I used another handy Flex component, the `RichTextEditor`. This component lets users to enter, edit, and format text. When the text is altered (for example recolored, size changed), the component stores the formatted text as HTML text, which makes it easy to export it. I slightly changed it, initially the text toolbar is on the bottom, I moved it to the top (users are more familiar with this layout), and added two additional buttons. One for saving the text and one for undoing the changes, and revert the text to the last saved state. Note that after pressing Save the Undo button cannot restore the initial state.

7 Testing

The purpose of the project was, to create an application with high compatibility while maintaining performance, and providing an engaging user experience. These are going to be the main fields during testing.

7.1 Compatibility

This field has been covered in some extent in the specification, because the developer has to think ahead when starting a project using a web-based technology. This field is constantly changing, and can be hard to keep the software compatible, even if the providers of the technology are trying to maintain a high level of backward compatibility. Several errors were detected end solved during this phase of the testing, below are the most significant ones.

As the worldwide ubiquity of Adobe Flash Player is above 99 % the vast majority of the visitors are covered. [10] The application was developed and tested in version 10.2. It can not be guranteed, that it will be functioning correctly in older versions, so if someone has an older version, or another problem is obstructing Flash Player to run the application, the application is displaying an error message. This eliminates the problems originated from wrong environment conditions.

In the latest version of Flash Player, Adobe improved security in many aspects. In order to allow Flash to handle data across multiple domains, some security restrictions have to be set in a configuration file called crossdomain.xml, and placed in the root folder of the server.

Different web browsers can have different behavior in some scenarios, it is necessary to test the application in the most used ones. The application uses HTTP requests for attaining the data files. Some browsers handle these requests differently, and have various caching strategies. Firefox by Mozilla has the most effective caching abilities, what caused several problems for the application, such as reloading the map after it has been changed. Firefox was forcing to use the data loaded into the cache. This issue has been eliminated, by adding a unique time variable to every request to bypass the usage of the cache. The application has been also tested in Microsoft Internet Explorer, and Google Chrome.

7.2 Performance

During the development it was an important aspect to maintain good performance, and keep the resource demand as low as possible. This applies on the data transmission, as well as the displaying engine. For the final testing and optimization different laptop and personal computer configurations were used: a smaller netbook, with low performance, an average notebook, with decent performance, and a personal computer with quite powerful processing capabilities.

During the testing, average resource demand was documented. All the configurations were able to run it, and provide a smooth operation. In the early stages there has been some heavy memory

usage, but it has been reduced successfully. Stress tests were included into the final testing session, and the weakest configuration started to function poorly when increasing the nodes above a certain number. This behavior was expected, and can be attributed to the system, not to the application. It can be asserted, that for regular usage, any modern computer is satisfactory.

7.3 User experience

After deploying the application, the next stage was to introduce it to a bigger audience, and get feedback from the users. In the first round, it has been presented to a smaller set of people with programming skills, then to a group of 30-40 people, and the results were examined altogether, as well as in comparison. This concept has been used, because users with informatics knowledge are more likely to reveal functionality flaws and help in the process of fixing them. Then a more stable application was given to the regular users, which could concentrate on the experience itself.

For feedback an online questionnaire was used, which was focusing on the subjective opinions. They're impressions can be summarized with the followings:

+ Nice design, a good overall feeling
+ Exciting map movement
+ Tooltips showing the outline
+ Tabbed text editing
+ Fast, slick response

– Missing password memorization
– No auto fit into view feature
– No folder creation for grouping maps

7.4 Evaluation

As the testing results indicate, the primary goals concerning the usability were fulfilled. The created application can be improved in many ways, but the users were pleased, and in our case, that was the most important aspect. There are just a few, if none limitations to the system requirements. These facts indicate that the application would do well also in commercial use.

8 Conclusion

This thesis was intended to summarize the realization of an application in web-based platform as well as to give valuable information from the field including available technologies, and existing applications. It pointed out the importance and the benefits of using graphs, and that is convenient to use the newest software technologies to integrate graphs into applications. The program itself went through the standard proposal and implementation steps, using a cross-platform development framework and several third-party software components. The final product meets the expectations, although it slightly differs from the sketch up during the proposal. The fact, that external components were used meant that other elements had to be altered to completely integrate them into the system.

In comparison with the existing tools, it implements more supplements, let's just mention the organic-like movement or the integrated text editor. As the testing indicates, the application could be improved in more ways. For instance, the graph drawing algorithm could be improved, to optimize the distribution of the nodes among the view-space. Besides of that the user comfort could be emended by remembering login information, implementing view styles. It would be also able to implement work-groups, as already mentioned in the specification.

I have benefited greatly from developing this application. I have acquired both theoretical and practical knowledge in creation of Rich Internet Applications, particularly with the tools of Adobe Flex and ActionScript. In addition I have revived my knowledge about informational systems, databases and scripting languages. Also I have become familiar with the method of mind mapping, and will definitely use it in my further studies, but also in practical life.

The aim of this project was, to create a visually attractive, easy-to-use application, which gets the attention of the visitors, and inspires them to create mind maps, what can positively affect they creativity and productivity. Considering the feedback from the test subjects, the aim has been fulfilled.

References

[1] J. M. Harris, J. L. Hirst, and M. J. Mossinghoff, *Combinatorics and Graph Theory*. Springer, 2008.

[2] G. Y. Katona and A. Recski, *Gráfelmélet és algoritmuselmélet*. 1994

[3] J.M. Burke, *Split-brain research: the philosophy behind the science*, Harvard University, 1992.

[4] S.P.Springer and G. Detsch, *Left Brain, Right Brain: Perspectives from Cognitive Neuroscience*, W H Freeman & Co, 1997

[5] T. Buzan, B. Buzan, *The mind map book*, Pearson Education, 2006.

[6] "ThinkBuzan - Support - Mind Map Gallery." [Online]. Available: http://www.thinkbuzan.com/it/support/mindmapgallery. [Accessed: 1-May-2012].

[7] "Thoughts on Flash." [Online]. Available: http://www.apple.com/hotnews/thoughts-on-flash/. [Accessed: 1-May-2012].

[8] "Flash to Focus on PC Browsing and Mobile Apps; Adobe to More Aggressively Contribute to HTML5 (Adobe Featured Blogs)." [Online]. Available: http://blogs.adobe.com/conversations/2011/11/flash-focus.html. [Accessed: 1-May-2012].

[9] G. Lawton, „New Ways to Build Rich Internet Applications," Computer, col. 41, Aug. 2008, pp. 10-12

[10] "PC Penetration | Statistics | Adobe Flash Platform runtimes." [Online]. Available: http://www.adobe.com/products/flashplatformruntimes/statistics.html. [Accessed: 12-Jan-2012].

[11] A. C. Team and R. Chun, *Adobe Flash CS4 Professional Classroom in a Book*. Peachpit Press, 2009.

[12] "ActionScript 3.0 overview | Adobe Developer Connection." [Online]. Available: http://www.adobe.com/devnet/actionscript/articles/actionscript3_overview.html. [Accessed: 29-Jan-2012].

[13] M. David, *Html5: Designing Rich Internet Applications*. Focal Press, 2010.

[14] R. Hawkes, *Foundation Html5 Canvas: For Games and Entertainment*. Apress, 2011.

[15] I. Zoratti, „MySQL Security Best Practices," Jun 2006, pp. 183-198

[16] H.E. Williams and D.J. Lande, *Web database applications with PHP and MySQL*, O'Reilly Media, Inc.,2004.

[17] J. Lazar, *User-Centered Web Development*. Jones & Bartlett Learning, 2001.

[18] T.M.J. Fruchterman, E.M. Reingold, and U. Of I. At U.-C.D. of C. Science, *Graph Drawing by force-directed placement*, Citeseer, 1990.

[19] S.G. Kobourov and K. Wampler, „Non-Euclidean Spring Embedders" *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, Nov. 2005, pp. 757-767.

Appendix A: CD media

The attached CD media contains the followings:

- Source code
- Compiled application
- PHP scripts
- Sample data
- AsDoc documentation
- The text of this study
- ReadMe.txt with instructions how to deploy and use the application